A4-Q5: Audio Enhancing

In [1]:
```
import numpy as np
import matplotlib.pyplot as plt
import scipy.io.wavfile
from IPython.display import Audio
from numpy.fft import fft, ifft, fft2, ifft2, fftshift, ifftshift
from joSigProc import *
```

In [2]:
```
# Load the audio recording
Omega, f = scipy.io.wavfile.read('recording.wav')
Audio(f, rate=Omega)
```

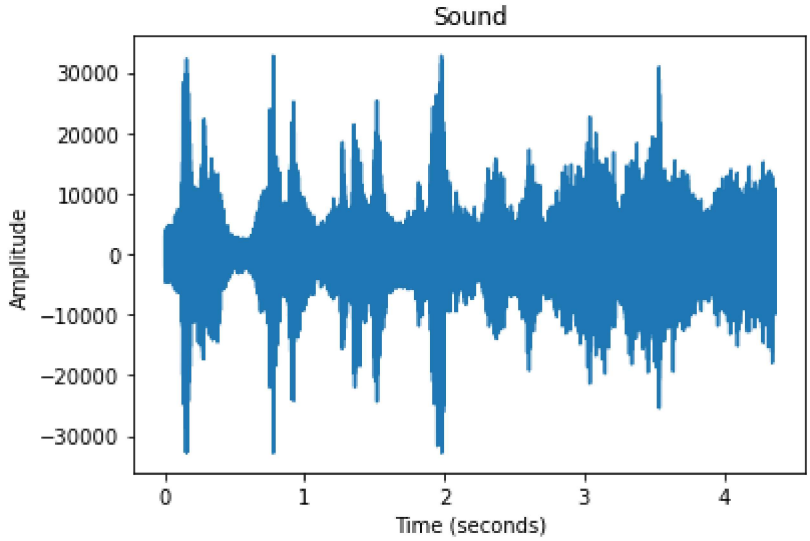
Out[2]:

0:00 / 0:04

In [3]:
```
# Some useful values
N = len(f)    # total number of samples
L = N / Omega # length of sound clip (in seconds)
t = np.arange(0,N) * L/N  # array of time stamps for samples
```

In [4]:
```
# Corresponding array of sampled frequencies
omega = np.fft.fftshift(np.arange(-N/2, N/2)) / L
```

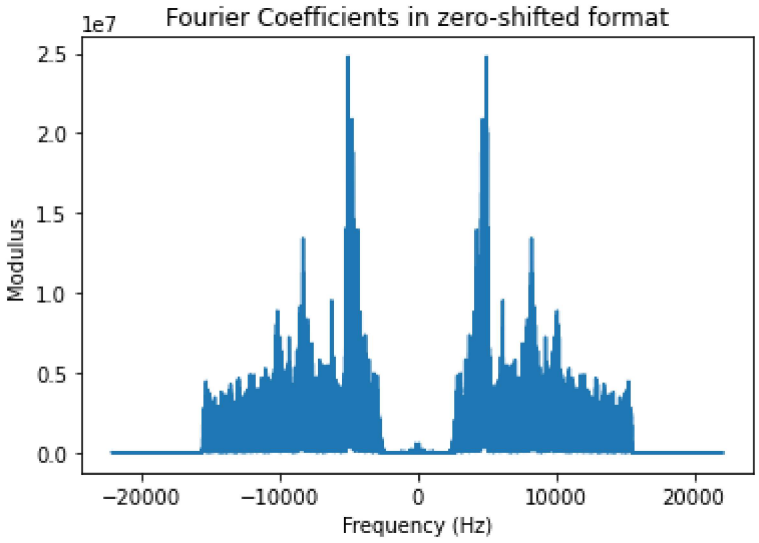
In [5]:
```
plt.plot(t, f)
plt.title('Sound')
plt.xlabel('Time (seconds)')
plt.ylabel('Amplitude');
```



(a)

In [6]:
```
F = fft(f)
omega = ShiftedFreqSamples(f, Omega)
plt.plot(omega, fftshift(abs(F)))
plt.title('Fourier Coefficients in zero-shifted format')
```

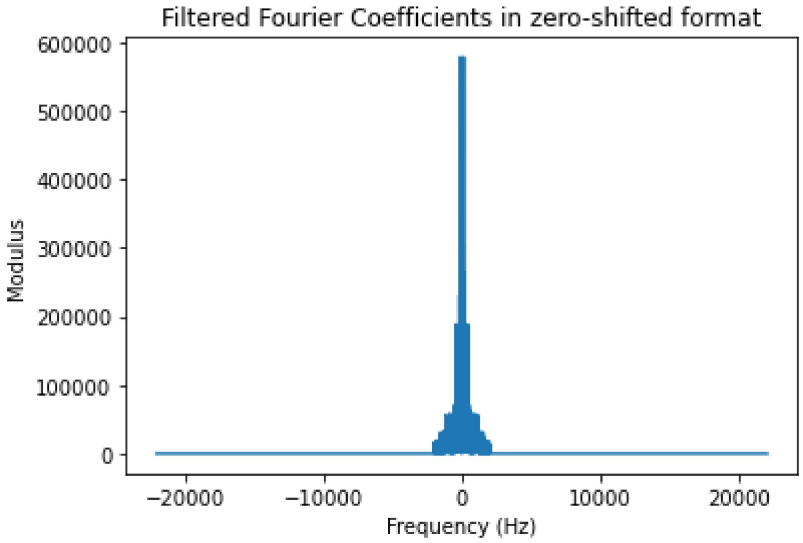
```
plt.xlabel('Frequency (Hz)')
plt.ylabel('Modulus');
```



(b)

In [7]:
```
T = 2000
G = F
G[abs(ifftshift(omega))>T] = 0.
```

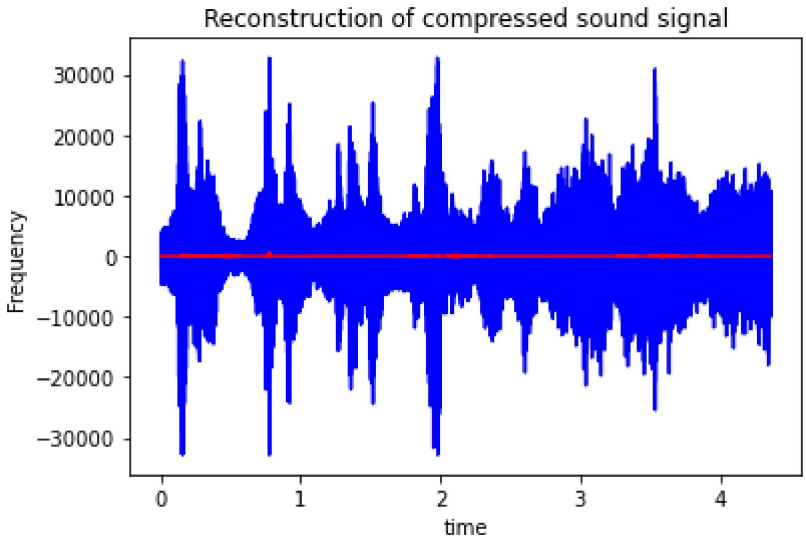
In [8]:
```
plt.plot(omega, fftshift(abs(G)))
plt.title('Filtered Fourier Coefficients in zero-shifted format')
plt.xlabel('Frequency (Hz)')
plt.ylabel('Modulus');
```



(c)

In [9]:
```
g = ifft(G)
plt.plot(t, f, 'b')
plt.plot(t, np.real(g), 'r')
plt.title('Reconstruction of compressed sound signal')
plt.xlabel('time')
plt.ylabel('Frequency')
```

Out[9]: Text(0, 0.5, 'Frequency')



In [10]:
```
Audio(g.real, rate=Omega)
```

Out[10]:

0:00 / 0:04

(d)

James Carver: Robert is dead. I killed him.